

HOMWORK 3

RyanKassab
kassab

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Late submissions may not be accepted. Please wrap your code and upload to a public GitHub repo, then attach the link below the instructions so that we can access it. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework.

https://github.com/kassabry/CS760_HW3

1 Questions (50 pts)

1. (9 pts) Explain whether each scenario is a classification or regression problem. And, provide the number of data points (n) and the number of features (p).
 - (a) (3 pts) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in predicting CEO salary with given factors.
Regression. There are 500 data points, each with 4 features: record profit, number of employees, industry and the CEO salary.
 - (b) (3 pts) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.
Classification. There are 20 data points, each with 13 variables and the outcome variable.
 - (c) (3 pts) We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.
Regression. There are 52 data points (weekly data for 2012), each with 4 features.
2. (6 pts) The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

X_1	X_2	X_3	Y
0	3	0	Red
2	0	0	Red
0	1	3	Red
0	1	2	Green
-1	0	1	Green
1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

- (a) (2 pts) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

<i>DataPoint</i>	Distance to Test Point
d_1	3
d_2	2
d_3	$\sqrt{10} = 3.1623$
d_4	$\sqrt{5} = 2.2361$
d_5	$\sqrt{2} = 1.414$
d_6	$\sqrt{3} = 1.7321$

- (b) (2 pts) What is our prediction with $K = 1$? Why?
 The prediction would be green because d_5 has the shortest distance to the test point, and the classification for that is green.
- (c) (2 pts) What is our prediction with $K = 3$? Why?
 The prediction would be red because out of the 3 nearest neighbors to the test point, 2 of them are classified as red, so the classification would be to the majority.
3. (12 pts) When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non-parametric approaches often perform poorly when p is large.
- (a) (2pts) Suppose that we have a set of observations, each with measurements on $p = 1$ feature, X . We assume that X is uniformly (evenly) distributed on $[0, 1]$. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with $X = 0.6$, we will use observations in the range $[0.55, 0.65]$. On average, what fraction of the available observations will we use to make the prediction?
 On average the available observations we will use would be 10%.
- (b) (2pts) Now suppose that we have a set of observations, each with measurements on $p = 2$ features, X_1 and X_2 . We assume that predict a test observation's response using only observations that (X_1, X_2) are uniformly distributed on $[0, 1] \times [0, 1]$. We wish to are within 10% of the range of X_1 and within 10% of the range of X_2 closest to that test observation. For instance, in order to predict the response for a test observation with $X_1 = 0.6$ and $X_2 = 0.35$, we will use observations in the range $[0.55, 0.65]$ for X_1 and in the range $[0.3, 0.4]$ for X_2 . On average, what fraction of the available observations will we use to make the prediction?
 Since there are two dimensions, the available observations used would be $(0.1)^2$ or 1%.
- (c) (2pts) Now suppose that we have a set of observations on $p = 100$ features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?
 Since there are 100 dimensions, the available observations used would be $(0.1)^{100}$.
- (d) (3pts) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations “near” any given test observation.
 Since the available training observations decrease exponentially as you increase the dimensions when you get to even 5 dimensions, the available training data near the point is only 0.0005% of the total data. Unless the data is extremely large or well clustered, KNN is unreliable to find a nearby neighbor that is near the point.
- (e) (3pts) Now suppose that we wish to make a prediction for a test observation by creating a p -dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For $p = 1, 2$, and 100, what is the length of each side of the hypercube? Comment on your answer.
 Each length of the hypercube would be $0.1 \times$ the length of the training observations. Hence why the possible available observations shrink exponentially. The volume of the hypercube would be $p_1 \times p_2 \times \dots \times p_{100}$.
4. (6 pts) Suppose you trained a classifier for a spam detection system. The prediction result on the test set is summarized in the following table.

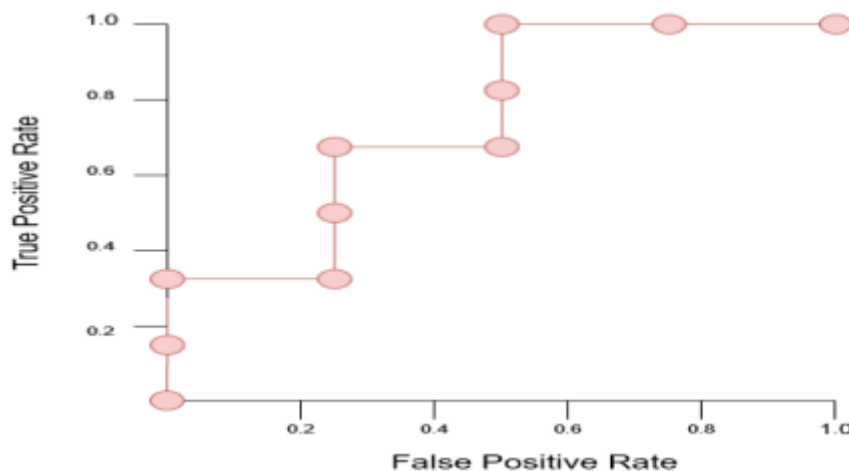
		Predicted class	
		Spam	not Spam
Actual class	Spam	8	2
	not Spam	16	974

Calculate

- (a) (2 pts) Accuracy [Accuracy = 0.982](#)
 (b) (2 pts) Precision [Precision = 0.333](#)
 (c) (2 pts) Recall [Recall = 0.8](#)
5. (9pts) Again, suppose you trained a classifier for a spam filter. The prediction result on the test set is summarized in the following table. Here, "+" represents spam, and "-" means not spam.

Confidence positive	Correct class
0.95	+
0.85	+
0.8	-
0.7	+
0.55	+
0.45	-
0.4	+
0.3	+
0.2	-
0.1	-

- (a) (6pts) Draw a ROC curve based on the above table.



- (b) (3pts) (Real-world open question) Suppose you want to choose a threshold parameter so that mails with confidence positives above the threshold can be classified as spam. Which value will you choose? Justify your answer based on the ROC curve.
- [Realistically, I would use 0.55 confidence as the cutoff, as the TPR is 0.67, but the FPR is only 0.25. If you decrease it to 0.45, the TPR stays the same, but the FPR increases to 0.5. While it could be beneficial to decrease slightly more and get a TPR of 1.0, the FPR being 50% is a problem.](#)
6. (8 pts) In this problem, we will walk through a single step of the gradient descent algorithm for logistic regression. As a reminder,

$$f(x; \theta) = \sigma(\theta^\top x)$$

$$\text{Cross entropy loss } L(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$

$$\text{The single update step } \theta^{t+1} = \theta^t - \eta \nabla_{\theta} L(f(x; \theta), y)$$

- (a) (4 pts) Compute the first gradient $\nabla_{\theta} L(f(x; \theta), y)$.

$$\begin{aligned}
L(\hat{y}, y) &= -[y\sigma(\theta^\top x) + (1 - y)\log(1 - \sigma(\theta^\top x))] \\
\frac{\delta}{\delta \theta} L(\hat{y}, y) &= -\frac{\delta}{\delta \theta} y\sigma(\theta^\top x) + \frac{\delta}{\delta \theta} (1 - y)\log(1 - \sigma(\theta^\top x)) \\
&= \frac{\delta}{\delta \theta} (\sigma(\theta_T x)) \left(-\frac{y}{\sigma(\theta_T x)} - \frac{1-y}{1-\sigma(\theta_T x)} \right) \\
&= ((\sigma(\theta_T x))[1 - (\sigma(\theta_T x))]x) \left(\frac{-y + \sigma(\theta_T x)}{(\sigma(\theta_T x))(1-\sigma(\theta_T x))} \right) \\
&= [-y + \sigma(\theta_T x)]x
\end{aligned}$$

- (b) (4 pts) Now assume a two dimensional input. After including a bias parameter for the first dimension, we will have $\theta \in \mathbb{R}^3$.

Initial parameters : $\theta^0 = [0, 0, 0]$

Learning rate $\eta = 0.1$

data example : $x = [1, 3, 2], y = 1$

Compute the updated parameter vector θ^1 from the single update step.

$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta} L(f(x; \theta), y)$$

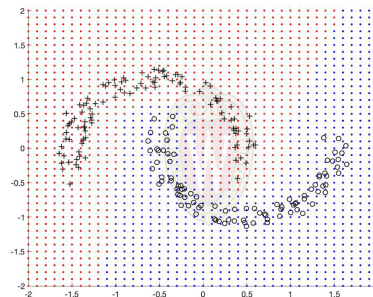
$$\theta^1 = \theta^0 - 0.1((0.5)x)$$

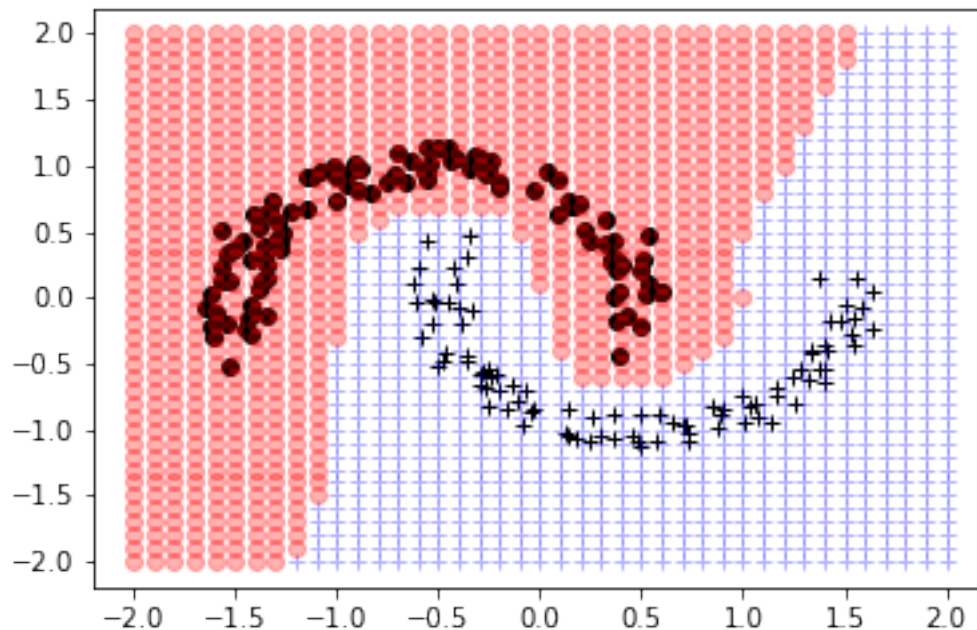
$$\theta^1 = [-0.05, -0.15, -0.1]$$

2 Programming (50 pts)

- (10 pts) Use the whole D2z.txt as training set. Use Euclidean distance (i.e. $A = I$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2]^2$. That is, you should produce test points whose first feature goes over $-2, -1.9, -1.8, \dots, 1.9, 2$, so does the second feature independent of the first feature. You should overlay the training set in the plot, just make sure we can tell which points are training, which are grid.

The expected figure looks like this.





Spam filter Now, we will use 'emails.csv' as our dataset. The description is as follows.

	Features																				Label
Email No.	the	to	ect	and	for	of	a	you	hou	in	...	connevey	jay	valued	lay	infrastructure	military	allowing	ff	dry	Prediction
Email 1	0	0	1	0	0	0	2	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Email 2	8	13	24	6	6	2	102	1	27	18	...	0	0	0	0	0	0	0	1	0	0
Email 3	0	0	1	0	0	0	8	0	0	4	...	0	0	0	0	0	0	0	0	0	0
Email 4	0	5	22	0	5	1	51	2	10	1	...	0	0	0	0	0	0	0	0	0	0
Email 5	7	6	17	1	5	2	57	0	9	3	...	0	0	0	0	0	0	0	1	0	0

- Task: spam detection
- The number of rows: 5000
- The number of features: 3000 (Word frequency in each email)
- The label (y) column name: 'Predictor'
- For a single training/test set split, use Email 1-4000 as the training set, Email 4001-5000 as the test set.
- For 5-fold cross validation, split dataset in the following way.
 - Fold 1, test set: Email 1-1000, training set: the rest (Email 1001-5000)
 - Fold 2, test set: Email 1000-2000, training set: the rest
 - Fold 3, test set: Email 2000-3000, training set: the rest
 - Fold 4, test set: Email 3000-4000, training set: the rest
 - Fold 5, test set: Email 4000-5000, training set: the rest

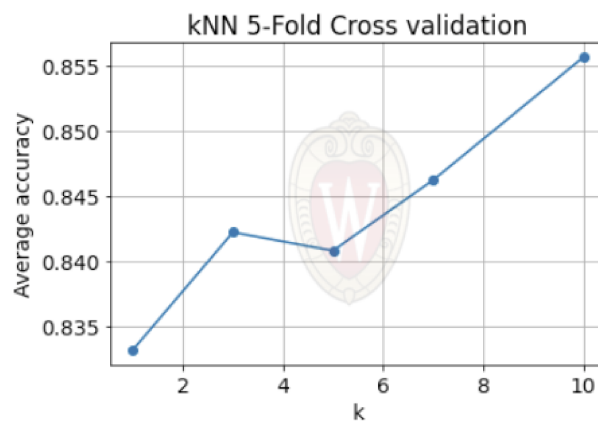
2. (8 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

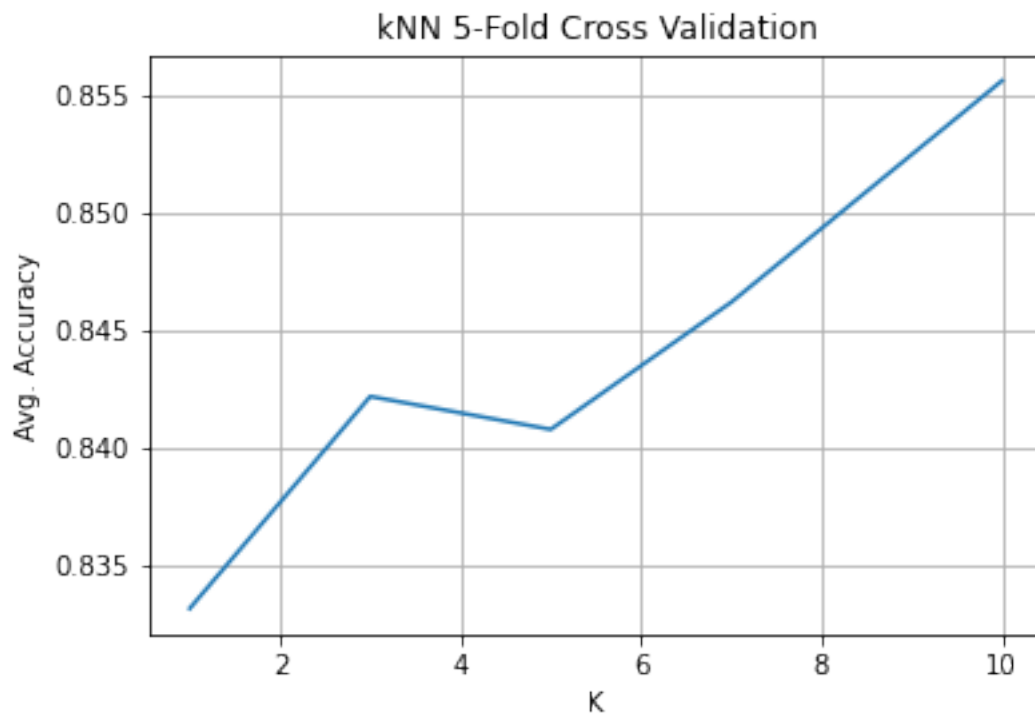
Fold	Accuracy	Precision	Recall
1	0.825	0.654	0.817
2	0.853	0.6857	0.866
3	0.862	0.7212	0.838
4	0.851	0.716	0.816
5	0.775	0.6057	0.758

3. (12 pts) Implement logistic regression (from scratch). Use gradient descent (refer to question 6 from part 1) to find the optimal parameters. You may need to tune your learning rate to find a good optimum. Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

Fold	Accuracy	Precision	Recall
1	0.723	0.5076	0.9298
2	0.783	0.875	0.253
3	0.864	0.885	0.598
4	0.747	0.9767	0.143
5	0.821	0.6799	0.784

4. (10 pts) Run 5-fold cross validation with kNN varying k ($k=1, 3, 5, 7, 10$). Plot the average accuracy versus k, and list the average accuracy of each case.
Expected figure looks like this.





5. (10 pts) Use a single training/test setting. Train kNN ($k=5$) and logistic regression on the training set, and draw ROC curves based on the test set.

Expected figure looks like this. Note that the logistic regression results may differ.



